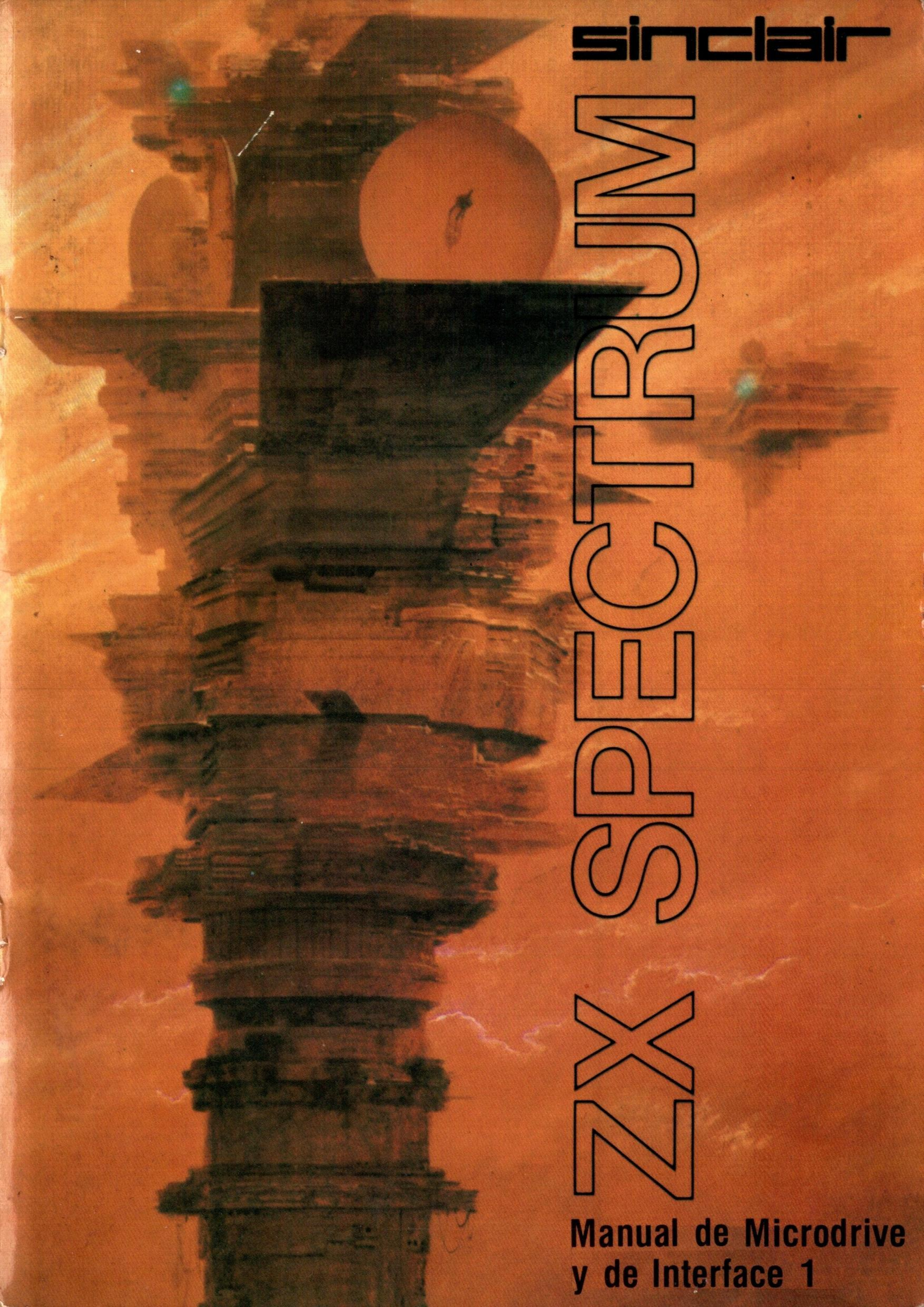
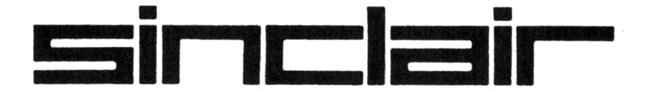

sinclair
ZX SPECTRUM
Manual de Microdrive
y de Interface 1



ZX INTERFACE 1
Y
ZX MICRODRIVE

Concebidos y preparados por
Cambridge Communication Limited

Nota
Los nombres **ZX Microdrive** y **ZX Interface 1** son marcas registradas de
Sinclair Research Limited
Stanhope Road
Camberley
Surrey
GU15 3PS
Tel.: Camberley (0276) 685311

Primera edición 1983

Ilustración de la portada por John Harris de Young Artists

EL ZX INTERFACE 1 ...

incorpora las tres funciones de controlar el Microdrive, la red de área local y el Interface RS232. Conéctelo a su Spectrum y podrá controlar hasta ocho Microdrives, comunicarse con otros ordenadores y utilizar una amplia gama de impresoras.

LA UNIDAD ZX MICRODRIVE ...

le proporciona acceso rápido a una memoria de gran capacidad. Cada Microdrive puede almacenar hasta 100 K bytes con el empleo de un solo cartucho intercambiable.

ESTE MANUAL ...

introduce la idea de emplear BASIC como un *sistema operativo* así como un *lenguaje de programación.* De esta forma, el BASIC puede utilizarse para establecer enlaces de comunicación, para enviar y recibir datos a través de los mismos y para manipular ficheros. Ello aumenta, en gran medida, sus posibilidades para realizar tareas en las que:

- se almacenen y recuperen datos,
- se envíen datos y programas a periféricos,
- se establezcan enlaces de comunicaciones con otros ordenadores Spectrum o con otra clase de ordenadores.

La utilización combinada del BASIC como sistema operativo y como lenguaje de programación le da potencia y flexibilidad, y también una facilidad de uso, que sólo puede encontrarse en muy pocos ordenadores más grandes.

Debe leerse los capítulos referentes a la unidad Microdrive aún cuando no la haya adquirido, puesto que introducen varios conceptos (acerca de *canales* y *corrientes* y sobre su empleo) que precisará comprender.

Este manual sólo debe leerse *después* de haberse familiarizado con el contenido del manual de introducción al ZX Spectrum y con los capítulos 1 a 22 del manual de programación en BASIC.

Indice

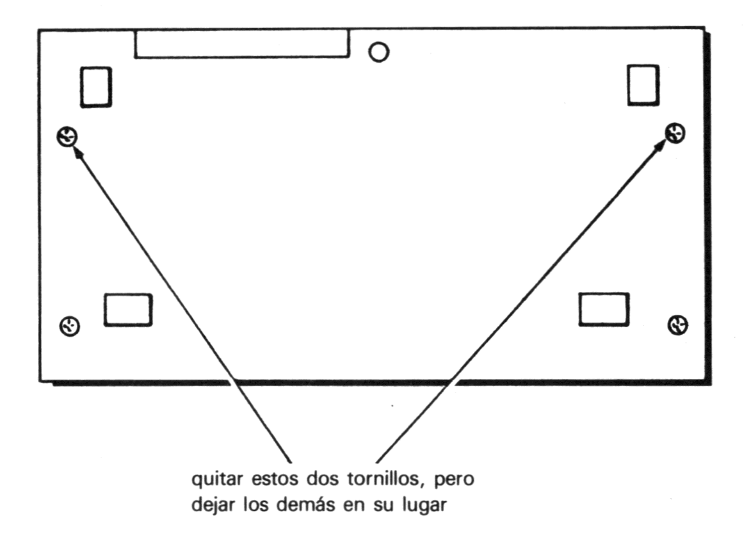
Instalación de su ZX Interface 1

Al desembalar el ZX Interface 1 encontrará:

- este manual,
- el propio Interface (con dos tornillos prisioneros en la parte inferior),
- un cable de cinta de unos 8 cm. de longitud (para su conexión a una unidad Microdrive),
- un cable con un enchufe de clavija en uno de los extremos (para instalación de una red).

Para conectar el Interface a su ordenador deberá:

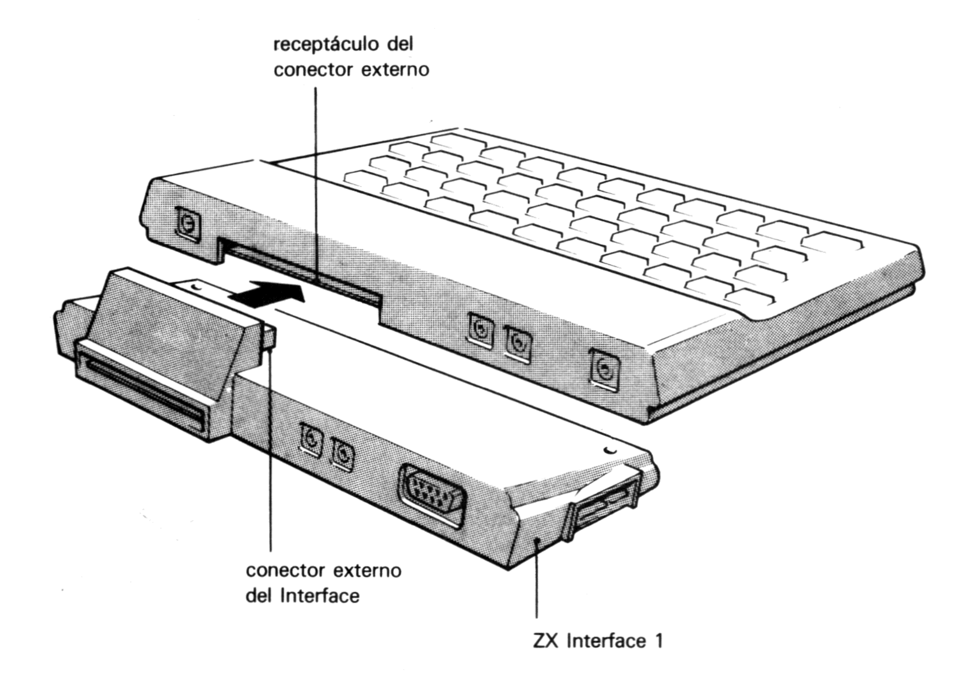
1. Desenchufar el ordenador.
2. Desconectar el ordenador de sus periféricos.
3. Con el empleo de un destornillador de estrella, quitar los dos tornillos en la parte inferior del ordenador, como se indica en la ilustración adjunta (solamente volverá a necesitar estos tornillos si ha de desconectar posteriormente el Interface del ordenador).



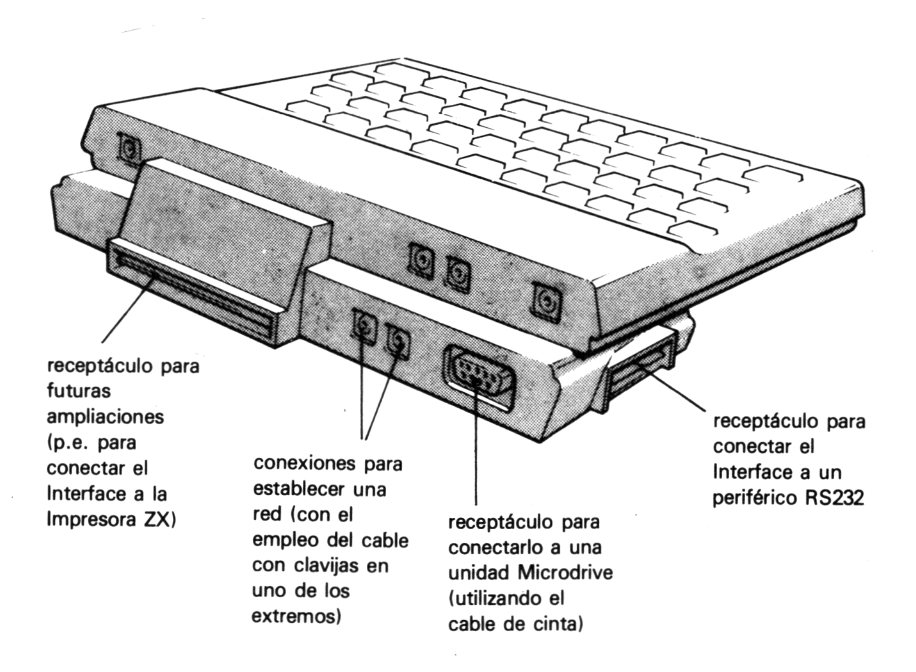
quitar estos dos tornillos, pero dejar los demás en su lugar

4. Introducir, con firmeza, el conector externo del Interface en el receptáculo correspondiente en la parte posterior del ordenador.
5. Atornillar el Interface a la parte inferior del ordenador con el empleo de los dos tornillos prisioneros suministrados.

Ahora puede volver a conectar el ordenador a sus periféricos y enchufarlo.



La ilustración adjunta muestra cuales son las diversas conexiones hembra en el Interface.



Ahora puede:

— instalar una unidad Microdrive ver página 9.

o

— instalar una red ver página 29.

o

— conectar el Interface a un periférico RS232 ver página 35.

Instalación de su Microdrive

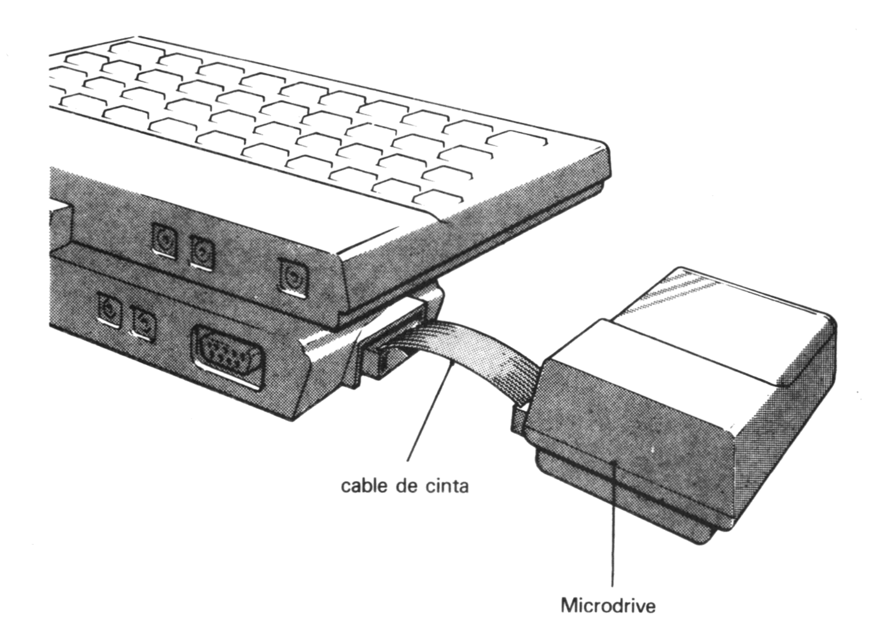
Al desembalar la unidad Microdrive habrá encontrado:

— la propia unidad Microdrive,

— un conector eléctrico (para conectar esta unidad con otra),

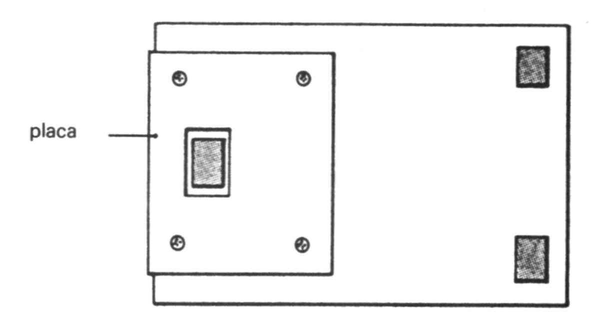
— un cartucho de demostración pregrabado.

Para conectar la unidad Microdrive al Interface, introducir con firmeza un extremo del cable de cinta (suministrado con el Interface) en el receptáculo correspondiente en la parte lateral del Interface. A continuación, insertar el otro extremo del cable en la conexión hembra en la parte lateral de la unidad Microdrive.

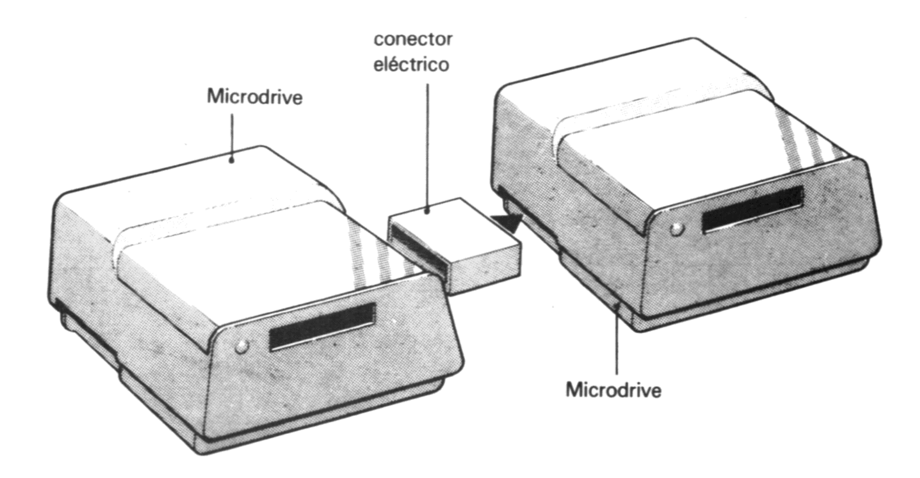


Si dispone de varias unidades Microdrive, la segunda puede conectarse a la primera como se indica a continuación:

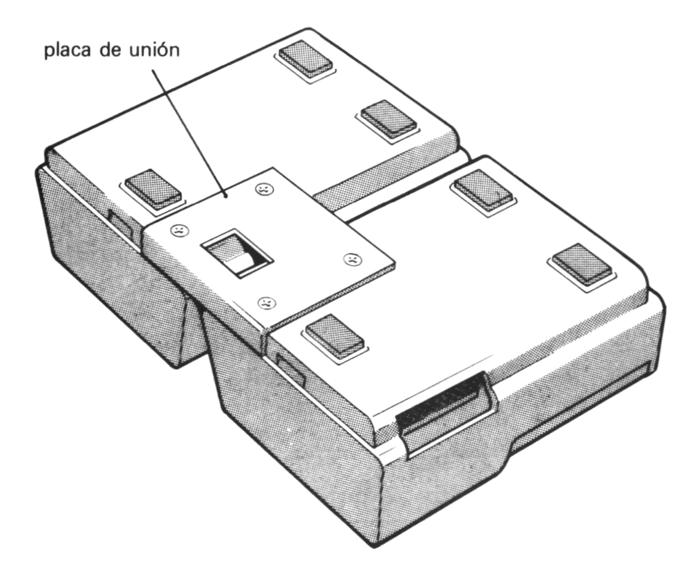
1. Desconectar el Microdrive del ordenador.
2. Dar la vuelta a los Microdrives. Descubrirá que cada uno tiene una placa atornillada a la parte inferior.



3. Extraer los tornillos de sujeción de las placas y retirar estas últimas.
4. Volver los Microdrives a su posición normal.
5. Introducir, con firmeza, uno de los conectores eléctricos en el receptáculo en la parte lateral de uno de sus Microdrives (como se muestra en la ilustración siguiente).
6. Empujar el segundo Microdrive contra el primero, de modo que el otro extremo del conector eléctrico se introduzca en el receptáculo en la parte lateral del segundo Microdrive.



7. Volver a dar la vuelta a los Microdrive, atornillar nuevamente una de las dos placas pero esta vez de modo que *sirva de enlace* de los dos Microdrives (el conector eléctrico, la placa y los tornillos restantes pueden conservarse para la posible conexión de otro Microdrive).



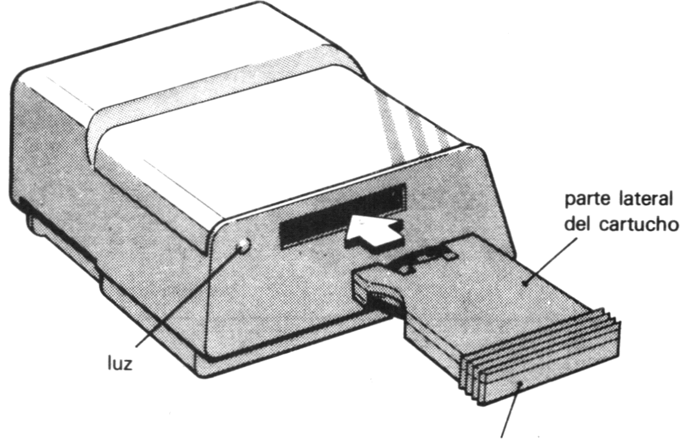
8. Volver los Microdrives a su posición normal y conectar el situado en el extremo derecho al ordenador.

> Tenga presente que el Microdrive más cercano al ordenador se conoce siempre como Microdrive 1 y el siguiente como Microdrive 2 y así sucesivamente.
>
> Obsérvese también que hay una luz en la parte frontal de cada Microdrive. Estará encendida siempre que esté funcionando el Microdrive.

Manipulación de cartuchos

Cada cartucho se suministra en una caja protectora y debe mantenerse siempre en su caja cuando no esté en uso.

Cuando saque un cartucho de su caja (teniendo cuidado en no tocar la cinta) verá que tiene una etiqueta en la parte superior y otra en la parte lateral.



Introduzca, con firmeza, el cartucho en la ranura existente en la parte frontal del Microdrive, cerciorándose de que queda hacia arriba la etiqueta del costado. De este modo, la parte superior del cartucho quedará sobresaliendo en 1/2 cm. aproximadamente. El Microdrive y el cartucho estarán preparados ya para su uso inmediato.

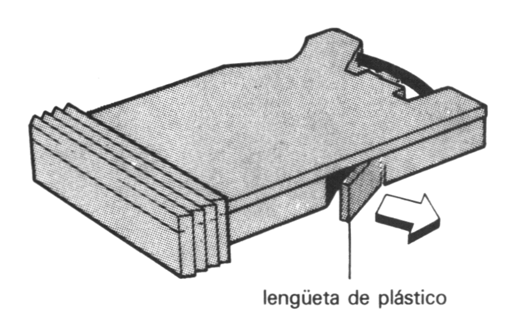
Cuando haya acabado de utilizar el cartucho, debe sacarlo cuidadosamente del Microdrive y volverlo a colocar inmediatamente en su caja. Pero recuerde

NO SACAR NUNCA EL CARTUCHO DE LA UNIDAD MICRODRIVE MIENTRAS SU LUZ ESTE ENCENDIDA.

NUNCA CONECTAR O DESCONECTAR LA ALIMENTACION MIENTRAS HAYA UN CARTUCHO EN EL MICRODRIVE.

Protección de los programas y datos almacenados en cartuchos

Si quiere proteger los programas y datos almacenados en un cartucho (es decir, evitar que se reescriban o se borren), debe quitar la lengüeta de plástico existente en el costado del cartucho, como se indica en la ilustración adjunta.



(Como en el caso de un cassette, puede invertir este efecto pegando una cinta adhesiva sobre el lugar en el que se ha quitado la lengüeta.)

La duración del cartucho

Los cartuchos del Microdrive no duran para siempre y a la larga, tendrá que sustituirlos. El síntoma del envejecimiento del cartucho es que el ordenador tardará cada vez más en encontrar un programa o fichero antes de que se cargue. Por ello resulta conveniente mantener copias de reserva de los programas importantes y de los ficheros en otro cartucho o en una cassette.

Puesta en marcha de su Microdrive

Funcionamiento automático

Ahora que ha instalado su ZX Interface 1 y su Microdrive, querrá conocer qué programas le están esperando en el cartucho de demostración. Para ello introduzca el cartucho en la unidad Microdrive (o si dispone de varias, en el Microdrive 1) y teclee:

NEW (y **ENTER**)

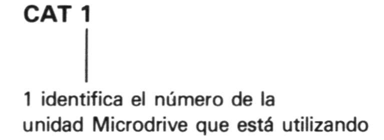
y a continuación:

RUN (y **ENTER**)

Con esta operación, se cargará y ejecutará automáticamente el primer programa en el cartucho. Cuando haya acabado de observar este programa, continúe leyendo.

El catálogo

Para averiguar qué otros programas se almacenan en el cartucho de demostración, introduzca la sentencia CATálogo:

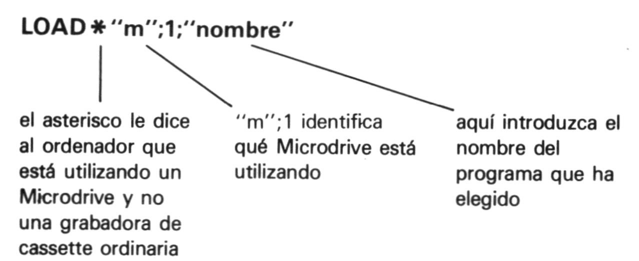


Transcurridos unos siete segundos, en la pantalla de televisión aparecerá:

- el nombre del cartucho,
- un catálogo alfabético de los nombres de todos los ficheros almacenados en el cartucho,
- la cantidad de memoria que queda en el cartucho (en kilobytes).

Carga de programas

La siguiente cosa a hacer es cargar **(LOAD)** cualquier programa que quisiere ver a continuación. Para ello elija primero su programa y luego, introduzca:



Transcurrido poco tiempo, aparecerá en la pantalla el mensaje OK (aunque no se indique el nombre del programa). Ahora ya puede ejecutar el programa.

Los programas y la unidad Microdrive

Grabación, verificación, carga y fusión de programas

En el manual de introducción del ZX Spectrum se le indicó cómo conservar **(SAVE)** programas en una cinta de cassette. La conservación de programas en cartuchos es igual de fácil. Volvemos a encontrarnos un programa llamado «Cuadrados», muy similar al del manual de introducción:

```
10 REM Cuadrados
20 FOR n=1 TO 10
30 PRINT  n,n*n
40 NEXT n
```

Para su conservación **(SAVE)** en una cinta de cassette, debe introducir:

SAVE "Cuadrados"

Para grabarlo en cartucho en el Microdrive 1, introduzca, en cambio:

SAVE*"m";1;"Cuadrados"

Transcurridos seis segundos, el contorno se hará parpadeante y se grabará el programa. Pruébelo y mire lo que obtiene.

(Los nombres de programas almacenados en un cartucho, como los de programas almacenados en una cinta de cassette, pueden tener hasta diez caracteres de longitud.)

Como esperaría, puede comprobar ahora que el programa se ha grabado introduciendo:

VERIFY*"m";1;"Cuadrados"

Ahora puede cargar **(LOAD)** el programa Cuadrados tecleando:

NEW (y **ENTER**)

seguido por:

LOAD*"m";1;"Cuadrados"

y luego:

ENTER

A continuación, para hacer que se ejecute automáticamente el programa, pruébelo introduciendo:

SAVE*"m";1;"Cuadrados 2" LINE 10

una vez finalizada la grabación, teclee:

NEW (y **ENTER**)

y luego:

LOAD*"m";1;"Cuadrados 2"

Los programas pueden también mezclarse con el empleo de un Microdrive. Para ello, introduzca:

NEW (y **ENTER**)

seguido por:

```
100 REM más Cuadrados
110 FOR n=11 TO 20
120 PRINT  n,n*n
130 NEXT n
```

E introduzca ahora:

MERGE*"m";1;"Cuadrados"

y luego:

ENTER

El programa Cuadrados se añadirá al listado.

En resumen, como probablemente ya habrá observado, toda la sintaxis utilizada dentro de un interface de cassette ordinario (explicado en el capítulo 20 del manual de programación en BASIC) se aplica también al Microdrive.

Borrado de programas

Supongamos que ha acabado con el programa Cuadrados. Para borrarlo, introduzca:

ERASE "m";1;"Cuadrados"

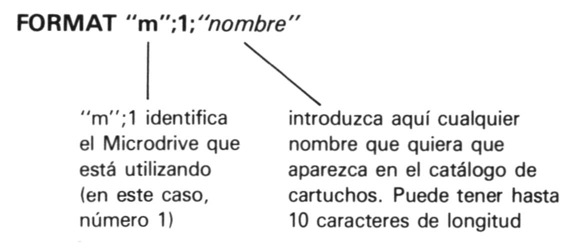
(como antes, "m";1 indica simplemente cuál es el Microdrive que está utilizando).

Durante la sentencia **ERASE** el contorno estará parpadeando.

Si produce una interrupción **(BREAK)**, mientras está grabando un programa, tendrá un fichero no cerrado en el cartucho. No puede cargar **(LOAD)** un fichero sin cerrar y un intento para hacerlo dará lugar al informe **file not found** (fichero no definido). La sentencia **ERASE** puede borrar un fichero no cerrado, pero tardará unos treinta segundos en hacer esta operación porque el ordenador comprueba varias veces el cartucho para cerciorarse de que el fichero no tiene ningún final.

Formateado y asignación de nombres a los cartuchos vírgenes

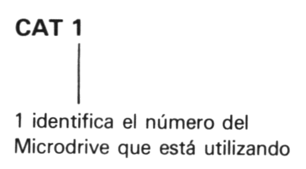
Si ha adquirido un cartucho virgen, antes de que pueda utilizarlo debe introducirlo en un Microdrive (digamos, en el Microdrive 1) y teclear:



El ordenador tarda unos treinta segundos en formatear un cartucho. Durante este tiempo, el contorno parpadeará primero, luego se borrará, luego volverá a parpadear y, finalmente, presentará el mensaje OK. Lo que el ordenador está haciendo es identificar cualquier zona que no pueda escribirse ni leerse, marcándola para evitar que pueda realizarse una u otra operación.

El formateado nunca es necesario que se repita. Tenga presente, a propósito, que cuando realice el formateado de un cartucho, se borrará cualquier información que contenga. Por ello, si ha adquirido varias unidades Microdrive, puede formatear **(FORMAT)** sus cartuchos de demostración de repuesto y utilizarlo como cartuchos vírgenes.

A continuación, teclee:



Transcurridos unos segundos, la pantalla de televisión presentará el nombre del cartucho y su capacidad. La capacidad variará de un cartucho a otro, pero nunca debe ser inferior a 85 kilobytes.

Establecimiento de la utilidad de ejecución automática

Anteriormente, utilizó la característica de ejecución automática en el cartucho de demostración. Si tiene un programa que suela utilizar, puede establecer en él la utilidad de ejecución automática, con miras a evitar el introducir repetidamente las sentencias **LOAD** y **RUN.** Las reglas a observar son las siguientes:

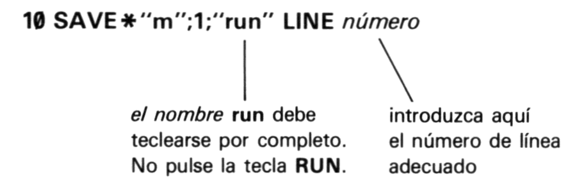
— el programa debe llamarse **run**;

— el cartucho debe utilizarse en Microdrive 1;

— la capacidad de ejecución automática debe utilizarse inmediatamente después del encendido o inmediatamente después de introducir **NEW.**

Por consiguiente, introduzca el programa requerido y a continuación:

10 SAVE*"m";1;"run" LINE *número*



Ahora, introduzca:

NEW (y **ENTER)**

y a continuación:

RUN (y **ENTER)**

introduzca la palabra clave y no el nombre del fichero

Sin embargo, tenga presente que la sentencia **MERGE** no actúa con cualquier programa grabado utilizando la sentencia **SAVE*** ... **LINE** ... (ello es para proteger dichos programas). Un intento de aplicar la función **MERGE** dará lugar al informe **'Merge' error.**

Datos, canales y corrientes

Como sabe, un programa es un conjunto de operaciones, que se ejecuta cuando teclea **RUN**. Por otra parte, los datos son cualquier serie de letras, números o símbolos sobre los que podría actuar un programa. Un ejemplo es el de los números 1 a 1Ø con sus cuadrados.

Los datos pueden enviarse a, y desde, diversas partes de un sistema de ordenador. Dichas partes son conocidas como *canales*. Los canales a los que puede enviarse datos son:

— su pantalla,

— una impresora ZX Printer,

— un fichero de Microdrive para su almacenamiento,

— otro ordenador ZX Spectrum, si los dos ordenadores están en una red,

— el interface RS232 y desde allí, por ejemplo, a un modem o impresora.

Y los canales desde los cuales pueden llegar datos son:

— el teclado,

— un fichero Microdrive,

— otro ordenador ZX Spectrum, si los dos ordenadores están en una red

— el interface RS232, un modem o un terminal.

Los caminos a lo largo de los cuales fluyen los datos a, y desde estos canales se conocen como *corrientes.* En el sistema de ordenadores Spectrum, el número de estas corrientes se fija en dieciséis. Están numeradas de Ø a 15, y los *números de corriente* siempre van precedidos por el signo # .

Cuatro de estas corrientes están enlazadas ya a canales:

corriente # Ø corriente # 1	*salida* de datos a la parte inferior de la pantalla de televisión y *entrada* desde el teclado;
corriente # 2	*salidas* a la parte superior de la pantalla de televisión, pero no puede ser *entrada;*
corriente # 3	*salidas* a través de la impresora; pero no puede ser *entrada.*

Cualquier sentencia que produzca entrada, o salida, utiliza automáticamente una de estas corrientes. Por ejemplo, la sentencia **PRINT** utiliza la corriente # 2, y la sentencia **LPRINT** utiliza la corriente # 3. Por ello, si introduce:

PRINT "Este es un ordenador Spectrum"

es, en realidad, una notación abreviada de:

PRINT # 2; "Este es un ordenador Spectrum"

Pruebe a introducir ambas y mire lo que sucede.

No obstante, puede hacer que cualquier sentencia utilice una corriente diferente tecleando # y a continuación, el nuevo número de corriente. Pruébelo, introduciendo:

LPRINT # 2; "Este es un ordenador Spectrum"

En lugar de imprimirse por la impresora, esta línea aparece también en la pantalla.

Pero además de utilizar corrientes-canales *establecidos* puede crear otros por su cuenta. Las corrientes # 4 a # 15 están libres para este propósito y hay diversos *especificadores de canales* que le permiten indicar qué periférico requiere. Algunos de estos especificadores son:

"K" para el teclado

"S" para la pantalla

"P" para la impresora

(más adelante, se le introducirán los demás).

Obsérvese que **K, S** y **P** son todos ellos canales establecidos. Le exigen el empleo de comas (,) como separadores en las sentencias **OPEN #** . Pero con los demás canales, puede utilizar comas o puntos y comas (;).

Para crear sus propias corrientes-canales ha de emplear la sentencia **OPEN #** . Por ejemplo, introduzca:

1Ø OPEN # 4, "S"

Habrá abierto la corriente 4 y la enlazará al canal **S.** Ahora, introduzca:

2Ø PRINT # 4; "Este es un ordenador Spectrum"

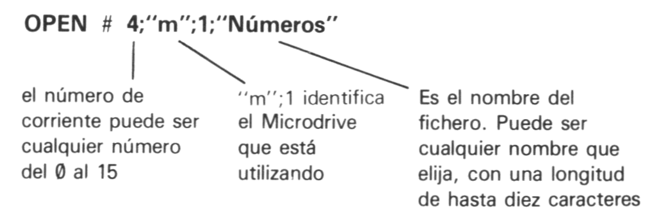
De nuevo, la línea aparecerá en la pantalla.

(No es recomendable aplicar **OPEN** a las corrientes Ø, 1 ó 2 pues los resultados son impredecibles.)

Los datos y la unidad Microdrive

Apertura y asignación de nombre a un fichero de datos

Cuando almacena información en un cartucho, ha de conservarla en un fichero. Asigne también al fichero un nombre de modo que pueda localizarlo más adelante. La sentencia que abre y da nombre a un fichero de datos toma siempre la misma forma. Por ejemplo, pruébelo introduciendo:



Esta sentencia realiza dos tareas bien distintas:

- establece un nuevo canal: "m";1;"Número",
- une este nuevo canal a la corriente # 4.

Esta operación habrá tenido una duración de unos siete segundos, durante cuyo intervalo el ordenador busca el cartucho para un fichero llamado "Números". Puesto que no hay ningún fichero llamado así, abre un fichero para escritura. (Sin embargo, si *hubiera* encontrado un fichero "Números", *lo* habría abierto para su lectura o habría encontrado un *programa* "Números" y entonces, emitiría el informe **Wrong file type,** que significa «tipo de fichero equivocado»).

Introducción de datos

Una vez que haya abierto un fichero, puede introducir datos. Supongamos que quiera almacenar los números 1 a 1Ø con sus cuadrados. Introduzca y ejecute lo siguiente:

```
1Ø FORn=1 TO 1Ø
2Ø PRINT # 4, n'n*n
3Ø NEXT n
```

Podría creer que todos los números se han almacenado ahora en el cartucho. Pero, en realidad, el ordenador no transfiere automáticamente nada al cartucho hasta que se haya acumulado una determinada cantidad y, entonces, transfiere todo a la vez. Este fenómeno se llama *buffering* (memorización intermedia). Un buffer (memoria intermedia) de Microdrive tiene una longitud de 512 bytes (o caracteres).

Para almacenar en el cartucho los datos que ha introducido, debe cerrar (CLOSE) el fichero. Hasta que dicha operación se realice, no podrá releer desde el fichero.

Cierre de un fichero

El cierre de un fichero garantiza que este último esté seguramente almacenado en el cartucho. También cierra el canal (en este caso, ''m'';1;''Números'') y deja a la corriente (en este caso, # 4) sin ningún canal unido. Para cerrar **(CLOSE)** un fichero, sólo precisa cerrar **(CLOSE)** la corriente correspondiente. Por consiguiente, introduzca:

```
CLOSE # 4
```

El contorno parpadeará para indicar que algo se está almacenando en el cartucho.

(Observe que, como en la sentencia **OPEN**, la sentencia **CLOSE** va seguida automáticamente por # .)

A propósito, no puede cerrar **(CLOSE)** las corrientes # 0, # 1, # 2, ó # 3. Si intenta hacerlo, las corrientes # **0** y # **1** darán por defecto el canal **K;** la corriente # **2** el canal **S** y la corriente # **3** el canal **P** (ver página 22).

Lectura de un fichero

Para la relectura desde el fichero ''Números'', ejecute lo siguiente:

```
10 OPEN # 4;"m";1;"Números"
20 FOR b=1 TO 10
30 INPUT # 4;m;n
40 PRINT "El cuadrado de ";m;" es ";n
50 NEXT b
60 CLOSE # 4
RUN
```

(dejar espacios aquí)

(Observe, en este punto, que puesto que ya existe el fichero ''Números'', el canal ''m'';1;''Números'' se abre para la entrada así pues, intentar la salida daría lugar a un error.)

También puede utilizar **INKEY$** para la relectura desde un fichero (siempre da el siguiente carácter en el fichero). Pruebe este programa:

```
10 OPEN # 11;"m";1;"listado"
20 LIST # 11
30 CLOSE # 11
40 OPEN # 12;"m";1;"listado"
50 PRINT INKEY$ # 12;
60 GO TO 50
```

Esto acabará con el informe End of file (final de fichero).

Observaciones sobre PRINT e INPUT

Puesto que las sentencias **PRINT** e **INPUT** están concebidas principalmente para su empleo con la pantalla y el teclado, debe tener cuidado cuando las utilice con ficheros.

separadores
la sentencia **PRINT** utiliza tres formas de separadores:

— el punto y coma (;) no imprime nada,

— la coma (,) le lleva al comienzo de la siguiente mitad de línea,

— el apóstrofe (') da una nueva línea (el código de **ENTER**).

La sentencia **INPUT** espera siempre que introduzca **ENTER** después de un número o de una cadena. En consecuencia, cuando esté enviando información a cualquier fichero del que espera una lectura **INPUT**, deberá:

— imprimir los elementos de datos de uno en uno, v.g.:

```
10 PRINT # 4;2
20 PRINT # 4;3
```

o bien

— separarlos con un apóstrofe, v.g.:

```
10 PRINT # 4;2'3
```

También debe tener cuidado cuando utilice separadores en una sentencia **INPUT**. Como sabe, **INPUT** puede imprimir en la mitad inferior de la pantalla cualquier cosa que pueda poner en una sentencia **PRINT**. Pero cuando realiza la función **INPUT** desde un fichero, este último sólo está abierto para la lectura. Por consiguiente, si incluye algo que se imprimiría cuando se está utilizando la pantalla, obtendrá el informe de error **Writing to a 'read' file.** Ello significa que los elementos de datos en la sentencia **INPUT** deben separarse con un punto y coma, v.g.:

```
10 INPUT # 4;a;b
```

También tenga cuidado cuando introduzca **(INPUT)** una cadena que contenga comillas (''), porque **INPUT** «creerá» que las comillas es el final de la cadena. La mejor forma de hacerlo es sustituir, por ejemplo:

```
10 INPUT # 4;a$
```

por:

```
10 INPUT # 4; LINE a$
```

cambio de corrientes

Las sentencias **PRINT** pueden contener también información para varias corrientes a la vez. El siguiente programa imprimirá «uno» en la pantalla, «dos» para un fichero de Microdrive denominado «dígitos» en el Microdrive 1; «tres» para la estación 1 en una red (ver capítulo 7) y «cuatro» para la siguiente línea en la pantalla.

```
10 OPEN # 4;"m";1;"dígitos"
20 OPEN # 5;"n";1
30 PRINT "uno"; # 4;"dos"; # 5;"tres" # 2"cuatro"
40 CLOSE # 4
50 CLOSE # 5
```

establecimiento de colores

Después de utilizar un canal que no sea la pantalla, puede encontrarse con que las sentencias **PAPER** e **INK** no tienen ningún efecto. Para subsanar esta anomalía, introduzca:

```
PRINT;
```

antes de establecer **PAPER** e **INK.**

Para más conocimientos sobre **PRINT** e **INPUT**, ver el capítulo 15 del manual de programación en BASIC.

Lectura del catálogo de ficheros

Cuando establezca ficheros en un cartucho, se catalogarán de forma automática. En consecuencia, cuando quiera averiguar qué ficheros contiene un cartucho, sólo necesitará insertar el cartucho en una unidad Microdrive e introducir la sentencia CATálogo. Por ejemplo, introduzca:

CAT 1

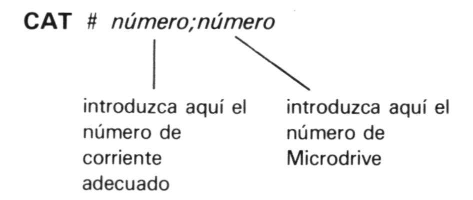
La pantalla de televisión presentará:

— el nombre del cartucho,

— los nombres de los ficheros,

— la cantidad de memoria que queda en el cartucho (kilobytes).

Asimismo, puede enviar la salida de **CAT** a una corriente introduciendo:



Ello le permite enviar el catálogo a una impresora, o a un fichero, de modo que un programa pueda utilizarlo.

Protección de un fichero

Si no quiere que un nombre de fichero aparezca en el catálogo, puede protegerlo dándole un nombre que comience con el carácter cuyo código (**CODE**) es 0. Introduzca lo siguiente:

```
10 OPEN # 4,"m";1;CHR$0+"Resultados"
20 FOR n=1 TO 15
30 PRINT # 4;n,n*n
40 NEXT N
50 CLOSE # 4
```

Ahora, introduzca:

CAT 1

No parecerá el nombre del fichero. Por consiguiente, siempre que cree un fichero protegido, recuerde tomar nota de su nombre en alguna parte, por si lo olvidara más tarde.

Extensión de un fichero

Supongamos que quiere ampliar el fichero "Números" para incluir los cuadrados de los números 1 a 2Ø, en lugar de 1 a 1Ø solamente. No puede volver a abrir un fichero para escritura, por lo que tendrá que:

— hacer una nueva versión con un nombre diferente;

— transferir el fichero antiguo a la nueva versión;

— añadir los datos suplementarios;

— borrar el fichero antiguo.

Veamos cómo hacerlo.

En primer lugar, ejecute **(RUN)** este programa:

```
 1Ø OPEN # 4;"m";1;"Números": REM para lectura
 2Ø OPEN # 5;"m";1;"Números 1": REM para escritura
 3Ø FOR f=1 TO 1Ø
 4Ø INPUT # 4;m;n
 5Ø PRINT # 5;m'n
 6Ø NEXT f
 7Ø FOR n=11 TO 2Ø
 8Ø PRINT # 5;n'n*n
 9Ø NEXT n
1ØØ CLOSE # 4:CLOSE # 5
```

Ahora, para comprobar que ha conseguido dos ficheros, "Números" y "Números 1", introduzca:

CAT 1

A continuación, para borrar el fichero antiguo, introduzca:

ERASE "m";1;"Números"

Para comprobar que se ha borrado, introduzca:

CAT 1

El nombre de fichero "Números" habrá desaparecido del catálogo y su nuevo fichero, "Números 1", contiene ahora los números del 1 al 2Ø.

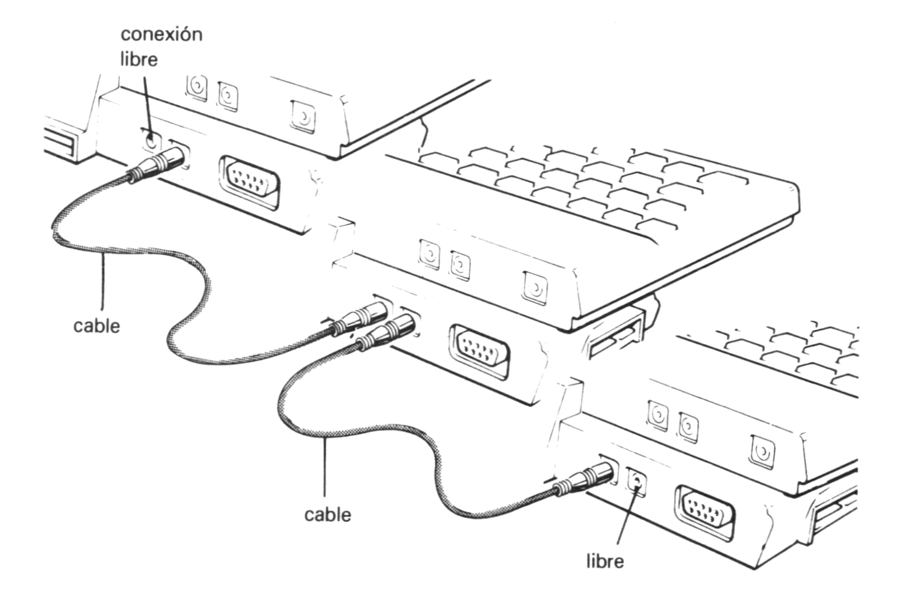
La red de area local

Establecimiento de una red

La red de área local, o *malla*, le permite a usted y a sus amigos disfrutar juntos de los juegos de ordenador e intercambiarse otros programas y datos. Ello significa que solamente uno del grupo necesitará teclear un programa. Una malla, o red, es útil sobre todo cuando solamente uno dispone de una unidad Microdrive.

Con el empleo del cable suministrado con cada Interface, puede enlazar desde dos hasta sesenta y cuatro ordenadores Spectrum, tal como se muestra a continuación.

No obstante, tenga presente que usted y sus amigos no deben formar un *bucle* de ordenadores; es decir, los ordenadores en cada extremo de la red nunca deben conectarse entre sí. Cada uno debe dejarse con una conexión de red libre.



NUNCA CONECTE O DESCONECTE UN SPECTRUM INCLUIDO EN UNA RED MIENTRAS LA COMUNICACION ESTE EN CURSO. No obstante, puede tener un Spectrum desconectado en la red y también puede conectar o desconectar Spectrums que estén en una red, *siempre que la comunicación no esté en curso.*

Cuando haya establecido una red, a cada ordenador (o *estación*) se le debe dar un número de identificación diferente. Por ello, antes de nada decida, con sus amigos, qué número ha de tener cada estación y luego, cada uno debe introducir:

FORMAT "n";número

introduzca aquí el número de estación que ha elegido

Circunstancialmente, tenga presente que si hay solamente dos ordenadores en una red, ambos pueden utilizar el mismo número de estación. Y, puesto que ambos ordenadores, cuando están conectados, se hacen automáticamente la estación 1, no hay ninguna necesidad, en absoluto, de utilizar la sentencia **FORMAT**.

Los programas y la red

Supongamos que tiene un amigo llamado Juan y que sus dos ordenadores están en una red. Su número de estación es, por ejemplo, 1 y el de Juan es 2.

Supongamos que quiere enviar a Juan este programa:

```
10 REM raíces
20 FOR n=1 TO 10
30 PRINT n,SQRn
40 NEXT n
```

Introduzca el programa, seguido por:

SAVE*"n";2

(observe que la red no utiliza nombres de programa).

Mientras tanto, Juan debe introducir:

FORMAT "n";2

Seguido por:

LOAD*"n";1

Juan tendrá ahora una copia del programa. Observe, no obstante, que mientras el ordenador está esperando la grabación **(SAVE)** o la carga **(LOAD)** de un programa en la red, el contorno **(BORDER)** de la pantalla se hace negro hasta que se envíe el programa. Su ordenador no hará el envío hasta que esté preparado el de Juan y este último esperará hasta que se envíe alguna cosa. Pruebe a introducir su línea de **SAVE** antes de que Juan introduzca su línea **LOAD** y viceversa.

Para verificar que Juan tiene ahora una copia de su programa, debe introducir:

VERIFY*"n";1

mientras que repite el envío del programa tecleando:

SAVE*"n";2

SAVE es, de hecho, la única sentencia que envía programas a través de la red. Las sentencias **LOAD, VERIFY** y **MERGE** son todas las formas de recibir programas.

El juego de la red en el Apéndice 1 es un buen ejemplo de cómo utilizar programas con una red.

Los datos y la red

Suponga que quiere ahora enviar a Juan algunos datos. La sentencia **OPEN # 4;"n";2** abre un canal a la estación 2 en la red ("n") y lo une a la corriente **# 4,** de modo que cuando efectúe la salida a través de la corriente **# 4,** su mensaje se introducirá en la red con una anotación que diga que procede de usted.

(Por el contrario, si introdujera **INPUT # 4;"n"2** su ordenador esperaría la información que le dirija Juan).

Ahora, introduzca este programa:

```
10 OPEN # 4;"n";2: REM para salida
20 INPUT a$: PRINT # 4;a$
30 CLOSE # 4
40 OPEN # 4;"n";2: REM para entrada
50 INPUT # 4;b$: PRINT b$
60 CLOSE # 4
70 GO TO 10
```

A continuación, introduzca:

SAVE*"n";2

Ahora, pida a Juan que introduzca:

FORMAT "n";2

y luego:

LOAD*"n";1

Ahora debe ejecutar su programa y Juan debe editar las líneas **10** y **40** del programa para hacerlas referirse a la estación 1 y no a la estación 2. A continuación, Juan debe introducir:

GO TO 40

Ahora estamos en condiciones de comenzar una conversación. Pero antes de que lo haga, hay tres cosas que debe conocer.

- Su salida a través de la corriente # *4* está *en la memoria intermedia,* es decir, no se aplica directamente a la red hasta que se haya acumulado una determinada cantidad de salida. Por ello siempre que acabe de realizar la función de salida debe cerrar **(CLOSE)** la corriente. Con ello se enviará la memoria intermedia (buffer) aunque no esté llena (las memorias intermedias tienen una longitud de 255 bytes o caracteres).

- Su salida está marcada como procedente específicamente *de usted,* de modo que si Juan está introduciendo o esperando la salida que venga de un canal de red diferente, se ignorará su mensaje. Si su mensaje *ha sido* ignorado, su pantalla no presentará el mensaje OK y el contorno **(BORDER)** de la pantalla se hará negro, hasta que Juan pida recibir el mensaje.

- Mientras que la sentencia **INPUT** espera simplemente que se envíe algo, la sentencia **INKEY$** puede utilizarse para la lectura de la red. Se producirá el reenvío con el primer byte de cualquier cosa que se haya enviado o que esté esperando que se envíe. (De cualquier otro modo, se tendrá el retorno de la cadena vacía.) Ello se conoce como «polling» (escrutinio por interrogación). De hecho, **INKEY$** actúa de la misma forma con la red como lo hace con el teclado.

El siguiente programa imprimirá algo que se le envíe por la estación 1:

```
10 OPEN # 8;"n";1
20 PRINT INKEY$ # 8;
30 GO TO 20
```

(Para más detalles sobre **INKEY$**, ver capítulo 18 del manual de programación en BASIC).

Difusión

Hay también un canal de red especial denominado de *difusión*, cuyo especificador es "n";Ø. Cuando realice una introducción desde este canal, captará cualquier mensaje que se esté difundiendo. Y cuando efectúe una salida, su mensaje puede ser leído por cualquiera que esté realizando una introducción desde el canal "n";Ø.

Por ejemplo, ello podría ser de gran utilidad en un centro escolar si cada alumno de una clase tuviera un ordenador Spectrum, pero solamente el profesor tuviera una unidad Microdrive.

Supongamos que el profesor quiere difundir un programa. En primer lugar, los alumnos deben introducir:

LOAD*"n";Ø

Con ello se dejará a los alumnos «esperando» recibir el programa. A continuación, el profesor debe grabar el programa introduciendo:

SAVE*"n";Ø

Como puede haber adivinado, las difusiones (a diferencia con los mensajes privados) se envían en seguida y no esperan a que otros ordenadores estén preparados para recibirlas.

A propósito, tenga presente que cuando envíe una difusión, el ordenador no le permite saber si alguien la ha recibido.

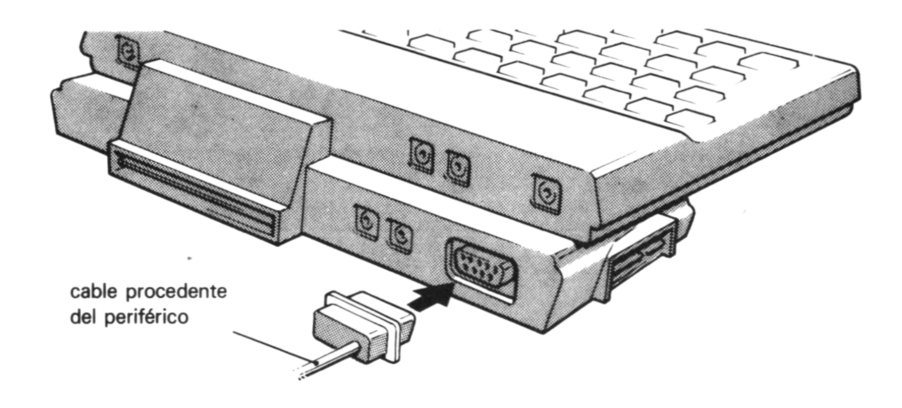
Asimismo, tenga en cuenta que **INKEY$** no realizará el «escrutinio» de interrogación del canal de difusión. Como **INPUT**, simplemente esperará que se envíe algo.

Empleo del Interface RS232

Conexión de los periféricos al Interface RS232

Como sabe, el conjunto de caracteres del Spectrum comprende *símbolos* simples (letras, dígitos, etc.) y entidades distintivas, «tokens» en inglés, tales como palabras clave, nombres de funciones, etc. Todos estos caracteres pueden enviarse y recibirse por el RS232 a, y desde, cualquier dispositivo en serie compatible; por ejemplo, una impresora, un modem u otro RS232 que esté conectado a una clase diferente de ordenador.

Para conectar cualquiera de estos periféricos al RS232, debe utilizar un cable con una clavija D de 9 vías en uno de los extremos y un enchufe adecuado para su periférico en el otro. Debe insertar la clavija en D en la parte trasera del RS232, como se indica en la ilustración adjunta (para detalles del cableado, ver el apéndice 4, página 49).



A continuación, antes de que pueda utilizarse el RS232 necesitará ajustar algunos de los controles en el periférico. Pueden incluir:

— el **avance de línea automático,** que **normalmente** debe estar **desactivado**. (El Spectrum da como salida un retorno del carro y un avance de línea en un canal **«t»**, pero sólo un retorno del carro en un canal **«b»**. Estos canales **t** y **b** se explican a continuación).

— la **activación/desactivación de la paridad,** que **debe estar desactivada.**

— el **número de bits,** puesto que el **Spectrum pone a la salida ocho bits.**

— el **número de bits de parada,** puesto que el **Spectrum emite un solo bit de parada.**

— la **velocidad de transmisión en baudios** (esto es, el número de bits por segundo). El Spectrum puede funcionar a todas las velocidades de transmisión en baudios, a saber:

  50, 110, 300, 600, 1.200, 2.400, 4.800, 9.600 y 19.200.

  Debe ajustar su periférico para trabajar con la más elevada velocidad de transmisión en baudios de la que sea capaz (verá más adelante cómo ajustar el Spectrum para utilizar la misma velocidad de transmisión en baudios).

Para una explicación de cómo realizar estos ajustes, ha de consultar el manual de instrucciones suministrado con su periférico.

Canales t y b

El RS232 utiliza dos canales diferentes: el canal **t** y el canal **b.**

El canal t

El canal **t** (o de texto) suele utilizarse para enviar listados. El canal **t** tiene el siguiente efecto sobre el juego de caracteres:

caracteres	
0-31	(los códigos de control) no se envían, con la salvedad del 13 (retorno del carro) que se envía como 13 seguido por 10 (retorno del carro y avance de línea).
32-127	(los códigos ASCII) se envían como normales.
128-164	(los caracteres gráficos) no se envían. En su lugar, se envía el mensaje (código 63).
165-255	(los «tokens») se expanden a los códigos ASCII 32-127.

En **INPUT** e **INKEY$**, el canal **t** espera caracteres de 7 bits, por lo que ignora el 8.° bit si lo hubiere.

Para utilizar el canal **t**, primero debe decirle al ordenador qué velocidad de transmisión en baudios ha de utilizarse. Para ello, introduzca:

10 FORMAT "t"; *«velocidad de transmisión»*

introduzca aquí la velocidad en baudios para la que ha ajustado su periférico

Ahora para abrir **(OPEN)** una corriente que le corresponda, introduzca:

```
20 OPEN # 3;"t"
30 LLIST
```

El contorno **(BORDER)** de la pantalla parpadeará y el listado se enviará al periférico. (Obsérvese que **LLIST** es la abreviatura de **LIST # 3**). Ahora, introduzca:

LPRINT "Este es un mensaje abreviado".

Este mensaje se enviará también al dispositivo.

Si su ZX Interface está conectado a un terminal o a un dispositivo RS232 capaz de emitir, podrá efectuar **INPUT** desde el terminal o el dispositivo. Introduzca lo siguiente:

```
10 FORMAT "t";baud rate
20 OPEN # 4;"t"
30 PRINT INKEY$ # 4;
40 GO TO 30
```

(*baud rate*: introduzca aquí la velocidad de transmisión en baudios a la que ha ajustado su periférico)

Ahora, cualquier cosa que teclee en el terminal o dispositivo aparecerá en su pantalla.

El canal b

El canal **b** (o *binario*) envía los códigos completos de 8 bits utilizados por el Spectrum y le permite enviar los códigos de control a impresoras, etc.

Con **INPUT** e **INKEY$**, el canal **b** espera caracteres de 8 bits.

SAVE y **LOAD** solamente actuarán con el canal **b**.

Si está utilizando modems para conectar su ordenador con otro Spectrum a través de una línea telefónica, o si está almacenando su programa en una clase diferente de ordenador que también tenga un RS232, puede querer grabar **(SAVE)** y cargar **(LOAD)** a través del RS232. Para hacerlo, introduzca:

FORMAT "b"; *baud rate*

(*baud rate*: introduzca aquí la velocidad de transmisión en baudios a la que ha ajustado su periférico)

Ahora, por ejemplo, introduzca:

```
10 REM cifras
20 FOR n=1 TO 10
30 PRINT n,n*RND
40 NEXT n
```

seguido por:

SAVE* "b"

La sentencia MOVE

Hasta ahora, solamente ha podido desplazar datos desde un programa a un canal o viceversa. Sin embargo, la sentencia MOVE le permite desplazar datos desde un canal a otro. Por ejemplo, para desplazar datos desde el teclado a la pantalla, introduzca:

10 MOVE # 1 TO # 2

y luego:

RUN

Cualquier cosa que introduzca por el teclado aparecerá en la pantalla. Sin embargo, descubrirá que cuando pulsa **BREAK** solamente imprime un espacio en la pantalla. Para escapar de esta «trampa», ha de pulsar **ENTER** hasta que la posición de impresión alcance la parte inferior de la pantalla. A continuación, cuando el ordenador le haga la pregunta **scroll?** debe pulsar **BREAK**. (A propósito, debe evitar desplazar datos desde el teclado a cualquier otra corriente puesto que puede suceder que sea incapaz de salir de dicho modo con **BREAK**.)

Con el empleo de la sentencia **MOVE**, también puede examinar los ficheros almacenados en cartuchos. Por ejemplo, establecer el fichero «Números» (ver página 23) y luego, examinar su contenido introduciendo:

10 MOVE "m";1;"Números" TO # 2

(Observe que no necesita **OPEN** ni **CLOSE** para abrir o cerrar el fichero por sí mismo. **MOVE** se encarga de hacerlo.)

Análogamente, para hacer una copia del fichero "Números" introduzca:

10 MOVE"m";1; "Números" TO "m";1; "Números 2"

En este caso, **MOVE** abre una corriente para lectura desde el fichero existente ("Números") y otra para escribir al nuevo fichero ("Números 2"). A continuación, lee los datos en "Números" y los escribe en "Números 2". Entonces, cierra las dos corrientes.

MOVE actuará con números de corrientes (tal como # **4**) y con especificadores de canales (tal como **"m";1;"Números")**. Sin embargo, tenga presente que las corrientes establecidas, # **1** a # **3**, *no* pueden ser especificadas por los especificadores de canales **K**, **S** o **P**.

Si dispone de una segunda unidad Microdrive, puede utilizar la sentencia **MOVE** para hacer copias de reserva de datos en otro cartucho. Introduzca:

10 MOVE "m";1;"Números" TO "m";2;"Números 2"

(Tenga en cuenta que **MOVE** solamente trabaja con ficheros de datos. Si quiere tener una copia de reserva de un programa, debe cargar **LOAD** el programa y grabarlo **SAVE**.)

La sentencia MOVE

Hasta ahora, solamente ha podido desplazar datos desde un programa a un canal o viceversa. Sin embargo, la sentencia MOVE le permite desplazar datos desde un canal a otro. Por ejemplo, para desplazar datos desde el teclado a la pantalla, introduzca:

10 MOVE # 1 TO # 2

y luego:

RUN

Cualquier cosa que introduzca por el teclado aparecerá en la pantalla. Sin embargo, descubrirá que cuando pulsa **BREAK** solamente imprime un espacio en la pantalla. Para escapar de esta «trampa», ha de pulsar **ENTER** hasta que la posición de impresión alcance la parte inferior de la pantalla. A continuación, cuando el ordenador le haga la pregunta **scroll?** debe pulsar **BREAK**. (A propósito, debe evitar desplazar datos desde el teclado a cualquier otra corriente puesto que puede suceder que sea incapaz de salir de dicho modo con **BREAK**.)

Con el empleo de la sentencia **MOVE**, también puede examinar los ficheros almacenados en cartuchos. Por ejemplo, establecer el fichero «Números» (ver página 23) y luego, examinar su contenido introduciendo:

10 MOVE "m";1;"Números" TO # 2

(Observe que no necesita **OPEN** ni **CLOSE** para abrir o cerrar el fichero por sí mismo. **MOVE** se encarga de hacerlo.)

Análogamente, para hacer una copia del fichero "Números" introduzca:

10 MOVE"m";1; "Números" TO "m";1; "Números 2"

En este caso, **MOVE** abre una corriente para lectura desde el fichero existente ("Números") y otra para escribir al nuevo fichero ("Números 2"). A continuación, lee los datos en "Números" y los escribe en "Números 2". Entonces, cierra las dos corrientes.

MOVE actuará con números de corrientes (tal como # **4**) y con especificadores de canales (tal como **"m";1;"Números")**. Sin embargo, tenga presente que las corrientes establecidas, # **1** a # **3**, *no* pueden ser especificadas por los especificadores de canales **K**, **S** o **P**.

Si dispone de una segunda unidad Microdrive, puede utilizar la sentencia **MOVE** para hacer copias de reserva de datos en otro cartucho. Introduzca:

10 MOVE "m";1;"Números" TO "m";2;"Números 2"

(Tenga en cuenta que **MOVE** solamente trabaja con ficheros de datos. Si quiere tener una copia de reserva de un programa, debe cargar **LOAD** el programa y grabarlo **SAVE**.)

Es prudente hacer copias de reserva de cualesquiera, datos o programas que quiera conservar.

La sentencia **MOVE** también puede enviar ficheros a una impresora. Por tanto, si tiene una impresora, introduzca:

```
10 MOVE "m";1;"Números" TO # 3
```

El programa Servidor de Impresora

Este programa permite a un Spectrum en una red controlar una impresora tipo RS232. Entonces, la impresora se puede utilizar por todos los demás ordenadores en la red. Ello es útil, por ejemplo, si un grupo de personas que utilizan Spectrums sólo disponen de una impresora de alta calidad y la desean compartir. El programa muestra también un empleo importante de la sentencia **MOVE**.

El ordenador con este programa debe ser siempre la estación 64 y siempre ha de entrar en contacto con la estación 62 (que es una estación establecedora de contactos). De este modo, el emisor utiliza temporalmente la estación 62 y le envía su número de estación real a partir del cual desplazará un fichero al canal **t**. Para establecer una estación de Servidor de Impresora, utilice este programa:

```
10 FORMAT "n";64
20 OPEN # 4;"n";62: INPUT # 4;a$: CLOSE # 4
30 MOVE "n"; CODE a$ TO "t"
40 OPEN # 4; "b": PRINT # 4;CHR$ 12: CLOSE # 4: RUN
```

(La línea 40 envía un avance de papel.)

El siguiente programa es el utilizado por el emisor. En primer lugar, la estación del emisor se ajusta temporalmente a la estación 62. Entonces, se envía el número de estación del emisor. A continuación, la estación del emisor vuelve por sí misma a su número de estación real. Finalmente, la línea 60 envía cualesquiera datos a imprimir (en este caso, el listado).

```
10 LET estación=número
```

(introduzca aquí el número de estación real del emisor)

```
20 FORMAT "n";62
30 OPEN # 4;"n";64: PRINT # 4;CHR$ estación: CLOSE # 4
40 FORMAT "n";estación
50 OPEN # 4;"n";64
60 LIST # 4
70 CLOSE # 4
```

El juego de la red

Hay una copia de este juego en el cartucho de demostración suministrado con la unidad Microdrive. Su nombre de fichero es **«net game»** (juego de la red). Este programa proporciona un buen ejemplo de cómo puede utilizarse la red. Partes de dicho programa podrían incluirse en programas elaborados por Vd.

El juego

Para jugar a este juego, usted y su adversario deben pensar cada uno en un número entre 1 y 100. El ganador del juego es quien adivina antes el número pensado por el otro. Por cada intento su ordenador le dirá cuán próximo se ha quedado del acierto.

El programa

La subrutina en la línea 500 decide quien es el usuario 1 y quien es el usuario 2. Ello se hace así para que, cuando se intercambien intentos de acertar, uno de los jugadores utilice la subrutina en la línea 1100 y el otro la subrutina en la línea 1200; de esta manera, el usuario 1 envía primero y el usuario 2 recibe primero.

Su ordenador decide quien es el usuario 1 enviando al ordenador de su adversario el mensaje «1» y quedando a la escucha. Si recibe la respuesta de «1», ello significa que el ordenador de su adversario se activó después del suyo. Por consiguiente, su ordenador envía un «2» al ordenador de su adversario y se hace asimismo el usuario 1. (Si, por el contrario, su ordenador recibe un «2» como respuesta, ello indica que el ordenador de su adversario estaba activado ya, y a la escucha, cuando su ordenador le envió el «1». En consecuencia, su ordenador se hará el usuario 2).

Si ambos ordenadores se ponen en marcha al mismo tiempo, se producirá una «colisión» y será necesario una rotura **(BREAK)** y comenzar de nuevo.

La parte fundamental del programa está relacionada con el intercambio de nombres, introduciendo el número secreto (que no se envía) y comparando luego los intentos de acertar. Primero se envían estos últimos y luego, las respuestas.

Las línea 190 en adelante detectan una victoria en el juego, la indican adecuadamente y luego, ofrecen otro juego.

```
10  GO SUB 500
20  PRINT; : BORDER 1 : PAPER 1 : INK 7 : CLS
30  PRINT "Juego de adivinar números" "Primero introduzca su
    número secreto, luego, adivine el de su adversario"
40  INPUT "¿Cuál es su nombre?"; a$
50  PRINT "Hola";a$
60  GO SUB 1000 + 100*usuario
70  PRINT "Está jugando"; b$
75  PRINT'a$,b$
80  INPUT "Piense un número (1 a 100)";a
90  IF a < 1 OR a > 100 OR a <> INT a THEN GO TO 80
130 INPUT "Haga un intento de acertar";b
140 LET a$=STR$b : GO SUB 1000 + 100*usuario
150 LET c=ABS (a—VAL b$)
160 IF c=0 THEN LET a$="Correcto": GO TO 170
161 IF c < 4 THEN LET a$="muy muy próximo": GO TO 170
162 IF c < 10 THEN LET a$="muy próximo": GO TO 170
163 IF c < 20 THEN LET a$="próximo": GO TO 170
164 IF c < 40 THEN LET a$="bastante próximo": GO TO 170
165 IF c < 60 THEN LET a$="no muy próximo": GO TO 170
166 LET a$="ni mucho menos"
170 GO SUB 1000 + 100*usuario
180 PRINT b$,a$
190 IF c=0 OR b$="Correcto" THEN GO TO 210
200 GO TO 130
210 IF b$="Correcto" THEN PRINT FLASH 1;"Victoria": FOR n=0 TO
    7 : BORDER n: BEEP.1,n: BEEP.1,n+16: NEXT n:GO TO 230
220 PRINT, "Derrota": FOR n=7 TO 0 STEP —1: BORDER n:
    BEEP.2,n: NEXT n
230 BORDER 1: INPUT "¿Otro juego? (S/N)";a$
240 IF a$="S" THE RUN 20
499 STOP
500 OPEN # 4;"n";0
510 PRINT # 4;"1"
520 CLOSE # 4
530 OPEN # 4;"n";0
540 INPUT # 4;a$
545 CLOSE # 4
550 LET a=4: IF a$="1" THEN OPEN # a;"n";0: PAUSE 5:
    PRINT # 4;"2": LET usuario=1
560 IF a$="2" THEN LENT usuario=2
```

```
570  CLOSE # 4
580  FORMAT "n";usuario: RETURN
1100 OPEN # 4;"n";3—usuario
1110 PRINT # 4;a$
1120 CLOSE # 4
1130 OPEN # 4;"n";3—usuario
1140 INPUT # 4;b$
1150 CLOSE # 4
1160 RETURN
1200 OPEN # 4;"n";3—usuario
1210 INPUT # 4;b$
1220 CLOSE # 4
1230 OPEN # 4;"n";3—usuario
1240 PRINT # 4;a$
1250 CLOSE # 4
1260 RETURN
```

Variables de sistema

Además de las variables dadas en el Capítulo 25 del manual de programación en BASIC del Spectrum, en el software del Microdrive, red de área local y RS232 se utilizan las variables siguientes:

Notas	*Dirección*	*Nombre*	*Contenido.*
X1	23734	FLAG 3	Indicadores.
X2	23735	VECTOR	Dirección utilizada para ampliar el intérprete BASIC.
X10	23737	SBRT	Subrutina de paginación ROM.
2	23747	BAUD	Número de dos bytes que determinan la velocidad de transmisión en baudios calculada como sigue: BAUD = (3500000/ (26 * veloc. transm. baud.)) −2. Puede utilizar esta expresión para establecer velocidades de transmisión no normalizadas.
1	23749	NTSTAT	Número de estación de red propia.
1	23750	IOBORD	Color de contorno **(BORDER)** utilizado durante E/S. Puede introducir **(POKE)** cualquier color que quiera.
N2	23751	SER_FL	Zona de trabajo de 2 bytes utilizada por RS232.
N2	23753	SECTOR	Zona de trabajo de 2 bytes utilizada por el Microdrive.
N2	23755	CHADD_	Almacenamiento temporal para CH_ADD.
1	23757	NTRESP	Almacén para código de respuesta de red.
1	23758	NTDEST	Comienzo de buffer (memoria intermedia) de red, contiene número de estación de destino 0-64.
1	23759	NTSRCE	Número de estación de fuente.
X2	23760	NTNUMB	Número de bloque de red 0-65535
N1	23762	NTTYPE	Código tipo encabezamiento.
X1	23763	NTLEN	Longitud de bloque de datos 0-255.
N1	23764	NTDCS	Suma de control de bloque de datos.
N1	23765	NTHCS	Suma de control de bloque encabezamientos.
N2	23766	D_STR1	Comienzo de especificador de fichero de 8 bytes. Número de unidad control 2 bytes 1-8.
N1	23768	S_STR1	Número de corriente 1-15.

Notas	*Dirección*	*Nombre*	*Contenido.*
N1	23769	L__STR1	Tipo de dispositivo... **«m»**, **«n»**, **«t»** o **«b»**.
N2	23770	N__STR1	Longitud de nombre de fichero.
N2	23772		Comienzo de nombre de fichero.
N8	23774	D__STR2	Segundo especificador de fichero de 8 bytes, utilizado por los comandos **MOVE** y **LOAD**.
N1	23782	HD__ØØ	Comienzo de espacio o zona de trabajo para **SAVE** (grabar), **LOAD** (cargar), **VERIFY** (verificar), y **MERGE** (fusionar) datos tipo código.
N2	23783	HD__ØB	Longitud de datos Ø-65535.
N2	23785	HD__ØD	Comienzo de datos Ø-65535.
N2	23787	HD__ØF	Longitud de programa Ø-65535.
N2	23789	HD__11	Número de línea.
1	23791	COPIES	Número de copias hechas por **SAVE**.
	23792		Comienzo de MAPs o CHANs del Microdrive.

ADVERTENCIAS

1. La apertura de una corriente para el Microdrive, o la red, requiere una determinada cantidad de memoria libre con la que crear un canal. Un canal del Microdrive es de 595 bytes y un canal de red es de 276 bytes. Estos canales se crearán por **OPEN #**, **MOVE** o por **SAVE/LOAD/VERIFY/MERGE**. Ello significa que un programa existente con espacio insuficiente por debajo de RAMTOP dará el informe **Out of memory** (fuera de memoria) para cualquiera de estas operaciones.

2. Otro efecto de la creación de estas memorias intermedias es desplazar el código máquina almacenado en una sentencia REM. Ello puede crear problemas. Por ello ponga siempre los programas de código máquina por encima de RAMTOP.

3. No es recomendable aplicar **BREAK** durante una operación de escritura del Microdrive (una durante la cual esté parpadeando el contorno de la pantalla), puesto que puede que acabe con un fichero sin cerrar. **ERASE** suprimirá los ficheros no cerrados, pero tardará unos treinta segundos en hacerlo, ya que el ordenador comprueba varias veces el cartucho para cerciorarse de que el fichero no tiene final.

Canal de Microdrive

Cada vez que se abre un fichero se crea una zona llamada CANAL en la área designada CHANS en el manual de programación en BASIC. Dicha área suele direccionarse por el registro IX en el software. El canal tiene una longitud de 595 bytes y contiene el buffer de 512 bytes.

El contenido del canal es el siguiente:

Ø		Dirección 8.
2		Dirección 8.
4		'M'.
5		Dirección de subrutina de salida en ROM.
7		Dirección de rutina de entrada en ROM.
9		Dirección 595.
11	CHBYTE	El contador de bytes en curso indica el siguiente byte a añadir o a quitar de la zona de datos en el margen de Ø-512 inclusive.
13	CHREC	Número de registro. Indica la posición del registro en un margen de fichero de Ø-255.
14	CHNAME	Nombre de fichero de 1Ø bytes con espacios a la derecha.
24	CHFLAG	Byte indicador, bit Ø puesto a "1"... abierto para escritura puesto a "O"... abierto para lectura bits 1-7... no utilizados
25	CHDRIV	Número de unidad Ø-7.
26	CHMAP	Dirección del MAP para esta unidad Microdrive.
28	CHMAP	12 bytes de preámbulo de encabezamiento... marca el comienzo del espacio de trabajo del encabezamiento.
4Ø	HDFLAG	Byte indicador, bit Ø puesto a "1" bits 1-7... no utilizados.
41	HDNUMB	Número de sector en el margen Ø-255.
42		No utilizado.
44	HDNAME	Nombre del cartucho y espacios a la derecha.
54	HDCHK	Suma de control de encabezamiento
55		12 bytes de preámbulo de bloque de datos... marca el comienzo del espacio de trabajo de datos.
67	RECFLG	Byte de indicador, bit Ø=Ø bit 2= no un fichero **PRINT** bits 3-7 no utilizados,
68	RECNUM	Número de este registro en el margen Ø-255.

69	RECLEN	Número de bytes de datos en este registro Ø-512.
71	RECNAM	Nombre de fichero con espacios a la derecha.
81	DESCHK	Suma de control de los 14 bytes precedentes.
82	CHDATA	512 bytes de datos.
594	DCHK	Suma de control de los 512 bytes precedentes.

MAP
Para cada Microdrive que contiene un fichero abierto hay una zona denominada MAP (mapa) creada en el área llamada «Microdrive maps» (mapas del Microdrive) en el manual de programación en BASIC. MAP contiene 32 bytes. Cada bit corresponde a un sector en la unidad Microdrive correspondiente. Si dicho sector contiene datos o si no es utilizable, el bit se pondrá a «1». Los bits están numerados como sigue: bit Ø byte Ø = sector Ø, bit 1 byte Ø = sector 1, bit Ø byte 1 = sector 8, y así sucesivamente.

Canal de red

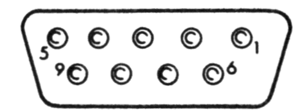
Cuando una corriente se abre a la red, se crea un canal en el área designada CHANS en el manual de programación en BASIC. Esta área suele direccionarse por el registro IX en el software. El canal tiene una longitud de 276 bytes y contiene el buffer de 255 bytes.

El contenido del canal se describe como sigue:

Ø		Dirección 8.
2		Dirección 8.
4		«N».
5		Dirección de subrutina de salida en ROM.
7		Dirección de subrutina de entrada en ROM.
9		Dirección 276.
11	NCIRIS	El número de estación de destino.
12	NCSELF	Número de estación de este SPECTRUM.
13	NCNUMB	El número de bloque.
15	NCTYPE	El código empaquetado... Ø'datos, 1EOF.
16	NCOBL	Número de bytes en el bloque de datos.
17	NCDCS	La suma de control de datos.
18	NCHCS	La suma de control de encabezamiento.
19	NCCUR	La posición del último caracter tomado de la memoria intermedia (buffer).
2Ø	NCIBL	El número de bytes en el buffer de entrada.
21	NCB	Un buffer de datos de 255 bytes.

Conexiones RS232

El conector RS232 está cableado como sigue:



1. Ninguna conexión.
2. Datos TX (entrada).
3. Datos RX (salida).
4. DTR (entrada) debe estar a nivel alto cuando esté preparada.
5. CTS (salida) debe estar a nivel alto cuando esté preparada.
6. n.c.
7. Masa (llevada abajo).
8. n.c.
9. + 9 V (llevada arriba).

Un cable RS232 está disponible a través de Sinclair Research y sirve para conectar el receptáculo en D de 9 vías con un enchufe macho en D de 25 vías (los enchufes hembra de 25 vías son comunes en los periféricos del RS232). Para los detalles de cómo obtener este cable, ha de consultarse el catálogo de periféricos y software que se incluye con el ZX Interface 1. Este cable tiene la disposición siguiente:

2. TX (transmisión) datos.
3. RX (recepción) datos.
5. CTS.
6. + 9 V (normalmente DSR).
7. Masa.
20. DTR.

Informes

Ahora que ha conectado un ZX Interface 1 a su ordenador, sus programas pueden producir informes que no se describen en el Apéndice B del manual de programación en BASIC. Dichos informes irán seguidos por el número de línea y el número de la sentencia en la que se detuvo el programa.

Estos nuevos informes se explican (en orden alfabético) a continuación.

Code error (Error de código)

Ha intentado cargar **(LOAD)** un bloque de código que es más grande que el área de destino especificada por su sentencia **LOAD**.

Drive 'write' protected (Unidad protegida contra la escritura)

Ha intentado escribir datos en un Microdrive que contiene un cartucho protegido (esto es, con la lengüeta de plástico de la parte lateral quitada).

File not found (Fichero no encontrado)

Significa que ha intentado cargar **(LOAD)** a partir de un fichero que no existe *o* que no se puede encontrar parte del fichero (ello se debería a que el fichero no se ha cerrado o a que el fichero se ha deteriorado por haber conectado o desconectado la alimentación mientras el cartucho estaba en el Microdrive).

Invalid device expression (Expresión de dispositivo inválido)

Se ha especificado un dispositivo de forma distinta a **s, p, k, m, t** o **b.** El mismo informe puede producirse si se utiliza un punto y coma, en lugar de una coma, con uno de los canales **s, p** o **k.**

Invalid drive number (Número de unidades no válido)

Un número de Microdrive se ha especificado fuera del margen de 1 a 8.

Invalid name (Nombre no válido)

Un nombre de fichero se ha especificado como una cadena vacía o una cadena con más de diez caracteres.

Invalid station number (Número de estación invalido)

Una estación de red se ha especificado fuera del margen de Ø a 64 (o fuera del margen 1 a 64 para las sentencias FORMAT).

Invalid stream number (Número de corriente no válido)

Se ha especificado una corriente fuera del margen Ø a 15.

Merge error (Error de fusión)

Ha intentado fusionar **(MERGE)** datos o código. Solamente puede hacerlo con programas. Este informe aparecerá también si ha intentado fusionar **(MERGE)** un programa grabado por la sentencia **SAVE... LINE...**

Microdrive full (Microdrive lleno)

Ha intentado escribir datos para un cartucho que no contiene ningún espacio libre. Por consiguiente, debe volver a ejecutar su programa u orden, con un cartucho que tenga espacio libre. Ello puede hacerse borrando ficheros antiguos del cartucho actual. Un fichero abierto para escritura en el cartucho lleno no se puede cerrar y debe borrarse. Sin embargo, esta operación tardará unos treinta segundos porque el ordenador comprueba varias veces el cartucho para cerciorarse de que el fichero no tiene final.

Microdrive not present (Microdrive no presente)

Ha intentado utilizar una unidad Microdrive que no está conectada a su ordenador o que no contiene un cartucho o que contiene un cartucho no formateado (codificado con alguna información que comunica formato y algunas otras características).

Missing baud rate (Falta la velocidad de transmisión en baudios)

No se ha especificado la velocidad de transmisión en baudios.

Missing drive number (Falta número de unidad)

No se ha especificado el número de Microdrive.

Missing name (Falta nombre)

No se ha especificado el nombre del fichero.

Missing station number (Falta número de estación)

No se ha especificado el número de estación de la red.

Program finished (Programa acabado)

Ha intentado ejecutar una línea más allá de cualquier línea existente. Este informe aparecerá si una instrucción GO TO va seguida por un número más allá de cualquier línea existente. También aparecerá si se teclea **RUN** sin un programa.

Reading a 'write' file (Intento de lectura en un fichero de «escritura»)

Ha intentado introducir datos en la memoria desde un fichero que no existe o que se ha abierto para escritura.

Stream already open (Corriente ya abierta)

Ha intentado abrir **(OPEN)** una corriente que se ha abierto ya para un nuevo canal **(m, n, t** o **b)**. La corriente sólo puede abrirse si se cierra primero.

Verification has failed (Ha fallado la verificación)

Un fichero conservado (o grabado) no coincide con el programa, los datos o el código existentes actualmente en el ordenador.

Writing to a 'read' file (Intento de escritura en un fichero de 'lectura')

Ha intentado enviar datos a un fichero existente. Este fichero debe borrarse si no se necesita. De cualquier otro modo, ha de utilizarse un nuevo fichero.

Wrong file type (Tipo de fichero equivocado)

Ha intentado aplicar **INPUT** o **MOVE** a un fichero grabado *o* **LOAD, VERIFY** o **MERGE** a un tipo de fichero **PRINT** *o* un fichero **CODE** o **DATA** se ha pretendido cargar **(LOAD)** como un programa (o viceversa).

Si está utilizando **INPUT,** habrá de emplear **LOAD**. Si está utilizando **LOAD,** deberá emplear las opciones de **CODE** o de **DATA** o utilizar **INPUT**.

El BASIC extendido

El ZX Interface 1 extiende el BASIC ya en el Spectrum. Las extensiones y adiciones se resumen a continuación.

Corrientes

Las corrientes se especifican como # n, en donde n es un número en el margen de 1 a 15. Las corrientes 1, 2 y 3 se suelen utilizar por BASIC. El carácter # es parte de la palabra clave para las sentencias **OPEN** # y **CLOSE** # .

Canales

Hay siete tipos de canal en el BASIC extendido; la pantalla **(s)**, el teclado **(k)**, la impresora ZX **(p)**, el Interface RS232 de texto **(t)**, el Interface RS232 binario **(b)**, la red **(n)** y la unidad Microdrive **(m)**.

Cada tipo de canal se especifica por su letra que puede ser mayúscula o minúscula. La red y el Microdrive requieren información suplementaria para especificar completamente el canal.

Un canal de red requiere un número de estación, por lo que un canal de red se especifica como **"n";X**, en donde x es un número de estación en el margen de Ø a 64.

Un canal de Microdrive requiere un número de Microdrive y un nombre de fichero, de modo que un canal de Microdrive se específica como **"m";y;"nombre"**, en donde **y** es el número de Microdrive en el margen de 1 a 8 y **"nombre"** es una cadena de caracteres entre 1 y 1Ø.

Sentencias

CAT y — Da una lista de todos los ficheros en el cartucho del Microdrive y. La lista se presenta en orden alfabético y va precedida por el nombre del cartucho y seguida por los kilobytes de capacidad restante.

CAT # **z;y** — Envía el catálogo del cartucho en el Microdrive y, tal como se describió anteriormente, a la corriente z.

CLOSE # **corriente** — Desconecta cualquier canal de la corriente especificada. Si hay algunos datos en la memoria intermedia, estos se transmitirán (en la red) o se registrarán (en el Microdrive).

Sentencias

ERASE "m";y;"nombre"	Borra el fichero con el nombre especificado a partir del cartucho en el Microdrive y.
FORMAT "m";y;"nombre"	Prepara un cartucho virgen de Microdrive en el Microdrive y para su empleo por BASIC. El nombre **"nombre"** se le da al cartucho y aparecerá en catálogos.
FORMAT "n";x	Hace x el número de estación de la red.
FORMAT "t";x **FORMAT "b";x**	Pone a x la velocidad de transmisión en baudios correspondiente al Interface RS232 (x debe elegirse entre uno de los valores normalizados de 50, 110, 300, 600, 1200, 2400, 4800, 9600, 19200).
INKEY$ # corriente	Proporciona un carácter único como una cadena si se dispone de un carácter y proporciona la cadena nula " " si no se dispone de ningún carácter a partir de la corriente. Esta instrucción sólo tiene significado si la corriente está enlazada a la red o al Interface RS232.
INPUT # Corriente;variable	Introduce la variable a partir de la corriente especificada. La corriente debe haberse abierto anteriormente para un canal de entrada. Es importante tener presente que cualquier elemento de datos de impresión en la sentencia **INPUT** será salida para la corriente. Ello sólo se suele requerir cuando se hace una introducción desde el teclado. También debe tenerse en cuenta que el separador "," corresponde a la salida de un carácter. La opción **LINE** está disponible como antes.
LOAD ✱ opciones de canal	Carga el programa, datos o código a partir del canal especificado. Solamente pueden utilizarse los canales **"b"**, **"n"** o **"m"**. Todas las opciones disponibles con **LOAD** lo están también con **LOAD ✱**.
MERGE ✱ opciones de canal	Lo mismo que con el **LOAD** anterior, con la salvedad de que no borra las antiguas variables o líneas de programa, excepto para dar paso a otras nuevas con el mismo nombre o número de línea.

Sentencias

MOVE fuente TO destino

Desplaza datos sobre la fuente al destino. La fuente y el destino pueden ser canales o números de corriente.

La orden termina solamente cuando una marca de final de fichero se encuentra en la fuente; ello sólo puede suceder si la fuente es una red o canal de Microdrive o bien una corriente enlazada.

Si la fuente o destino es un canal, se abrirá efectivamente primero y se cerrará después.

OPEN # corriente, canal

Enlaza el canal especificado con la corriente especificada para permitir la entrada o salida de BASIC a ese canal. La corriente debe cerrarse o abrirse previamente a **k, s** o **p.**

PRINT # corriente...

Salidas de secuencia de impresión (...) a la corriente especificada. Dicha corriente deberá haber sido abierta previamente a un canal de salida.

La secuencia de impresión tiene la misma sintaxis que antes y puede contener también entidades distintivas # .

SAVE*opciones canales

Conserva el programa, datos o código para el canal especificado. Solamente puede utilizarse los canales "**b**", "**n**" o "**m**".

Todas las opciones disponibles con **SAVE**, lo están también con **SAVE***.

VERIFY*opciones canales

Lo mismo que para el **LOAD** anterior, con la salvedad de que los datos no se cargan sino que se comparan con los datos ya existentes.

Indice

Este índice incluye las teclas en el teclado y cómo obtenerlas.

A

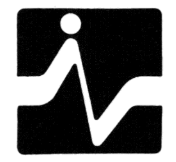
B

C

E

F

G



INVESTRONICA
Tomás Bretón, 21
MADRID-7 - SPAIN
Teléfs. 468 03 00 / 467 29 16
Télex: 23399IYCO E